C PROGRAMMING

Submitted by

Neena K

PROGRAM NO: 1

AIM: Simple program using formatted I/O.

ALGORITHM

Step 1 : start

Step 2 : read values of a and b from user

Step 3 : print the values of a and b

Step 4 : read value of n from user

Step 5 : print the value of n, right justified in the field of 6 columns

Step 6 : print the value of n, right justified in the field of 2 columns

Step 7 : print the value of n, left justified in the field of 6 columns

Step 8 : print the value of n, placing zeros before n in the field of 6 columns

Step 9 : read values of x and y from user Step 10 : read value of w from user

Step 11 : read value of p from user

Step 12 : print the values of x, is rounded to p decimal places and right justified in the field of w columns Step 13 : print the value of x in exponential form with default precision 6

Step 14 : print the value of y, is rounded to 2 decimal places and left justified in the field of 7 columns Step 15 : print the value of y in exponential form , is rounded to 2 decimal places and right justified in the field of 10 columns

Step 16 : read value of str from user

Step 17 : print the value of str in the field of 20 columns 3

Step 18 : print the first 10 characters of the string str in the field of 30 columns and right justified

Step 19 : print the first 5 characters of the string str

Step 20 : print the first 10 characters of the string str in the field of 20 columns and left justified

Step 21 : print the value of str

Step 22 : stop

PROGRAM

#include<stdio.h>

void main ()

{

int a,b,d,w,p;

float x;

double y;

char str[20];

printf("Formatted Input&Output\n");

printf("enter three integer numbers\n");

scanf("%d%\*d%d",&a,&b);

printf("value in a is%d\n value in b is%d\n",a,b);

printf("enter a 4 digit number\n");

scanf("%4d",&d);

printf("%6d\n",d);

printf("%2d\n",d);

printf("%-6d\n",d);

printf("%06d\n",d);

printf("Enter two floating point numbers\n");

scanf("%f%f",&x,&y);

printf("Enter the field width\n");

scanf("%d",&w);

printf("Enter the field width\n");

scanf("%d",&w);

printf("Enter the precision\n");

scanf("%d",&p);

printf("%\*.\*f\n",w,p,x);

printf("%e\n",x);

printf("%-7.2f\n",y);

printf("%10.2e\n",y);

printf("Enter a string\n");

scanf("%s",str);

printf("%20s\n",str);

printf("%30.10s\n",str);

printf("%5s\n",str);

printf("%-20s\n",str);

printf("%5s\n",str);

OUTPUT:

FORMATTED INPUT & OUTPUT

Enter three integer numbers

1

2

3

value in a is 1

Value in b is 3

Enter a 4 digit number

4526

4526

4526

4526

004526

Enter two floating point number

72.3

18.4

Enter the field width

4

Enter the precision

3

72.300

7.230000e+01

5 0.00

5.41e-315

Enter a string

malayalam

malayalam

malayala

malay

malayalam

malayalam

PRORAM NO: 2

AIM: Find the reverse of number using typedef

ALGORITHM

Step1: start

Step2: give a new name to int data type using typedef statement

Step3: read the value of number n,sum s,reverse rev

Step 4:Set rev=0 and s=0

Step 5: Check n>0

Step 6: Calculate s=n%10

Step 7: rev=rev\*10+s and n=n/10

Step 8: Display the reverse

Step 9: Stop

PROGRAM

#include<stdio.h>

void main()

{

typedef int number;

number n,s,rev;

printf("Enter a number\n");

scanf("%d",&n);

rev=0;

s=0;

while(n>0)

{

s=n%10;

rev=rev\*10+s;

n=n/10;

printf("Reverse is %d",rev);

}

}

OUTPUT

Enter a number

345

Reverse is 543

PROGRAM NO: 3

AIM: Find grade of a student using else if ladder.

ALGORITHM

Step 1 : start

Step 2 : read values of m1,m2and m3 from use

r Step 3 : Find average by add m1,m2 and m3 then divide by 3

Step 4: if average is greater than 90 then print A+ else go to step 5

Step 5 : if average greater than or equal to 80 and less than 90 print B+ else go to Step 6

Step 6 : if average is greater than or equal to 70 and less than 80 print C+ else go to step 7

Step 7 : if average is greater than or equal to 60 and less than 70 print D+ else go to step 8

Step 8 : Print „Fail „ and go to step 9

Step 9 : stop

PROGRAM

#include<stdio.h>

void main()

{

char n;

int m;

printf("Enter the name of the student");

scanf("%c",&n);

printf("Enter the mark");

scanf("%d",&m);

if(p>=90)

{

printf("A+");

}

else if(:>=80&&m<=90)

{

printf("A");

}

else if(m>=70&&m<=80)

{

printf("B");

}

else if(m>=60&&m<=70)

{

printf("C");

}

else if(m>=50&&m<=60)

{

printf("D");

}

else if(m<=39)

{

printf("Fail");

}

}

OUTPUT:

Enter three marks

38

49

36

Fail

Enter three marks

89

90

76

B

PROGRAM NO: 4

AIM: Find the corresponding numbers of Roman letters using switch case statement

ALGORITHM

Step 1: Start

Step 2: Read the value of r roman number

Step 3: check the value of r

Step 4: r=I Display 1

Step 5: r=V Display 5

Step 6: r=L Display 50

Step 7: r=C Display 100

Step 8: r=D Display 500

Step 9: r=M Display 1000

Step 10: r !=I or V or L or C or D or M Display Error

Step 11:Stop

PROGRAM

#include<stdio.h>

void main()

{

char r;

printf("Enter a roman letter\n");

scanf("%c",&r);

switch(r)

{

case 'I':

printf("1");

break;

case 'V':

printf("5");

break;

case 'L':

printf("50");

break;

case 'C':

printf("100");

break;

case 'D':

printf("500");

break;

case 'M':

printf("1000");

break;

default:

printf("Error");

}

}

OUTPUT:

Enter a roman letter

C

100

Enter a roman letter

P

Error

PROGRAM NO: 5

AIM: Find Armstrong numbers within a range using while loop.

ALGORITHM

Step 1 : start

Step 2 : read value of l from user

Step 3 : set i=l

Step 4 : Repeat while number less than or equal to l

Step 5 : set n=I and s=0

Step 6 : Repeat while n>0

Step 7 : calculate a=n%10

Step 8 : calculate p=d\*d\*d ,s=s+p and n=n/10.

Step 9 : check s==1 if it is true go to step 10

Step 10 : print value of i

Step 11 : increment I by 1 and go to step 4

Step 12 : stop

PROGRAM

#include<stdio.h>

void main()

{

int 1,n,d,i;

printf("enter the limit");

scanf("%d",&1);

printf("amstrong numbers are");

i=1;

while(i<=10)

{

int p,s;

n=i;

s=0;

while(n>0)

{

d=n%10;

p=d\*d\*d;

s=s+p;

n=n/10;

}

if(s==i)

{

printf("%d",i);

}

i++;

}

}

OUTPUT:

Enter a limit:

500

Amstrong numbers are:

1

153

370

371

407

PROGRAM NO: 6

AIM: Find the sum of n natural numbers using do while loop

ALGORITHM

Step 1: Start

Step 2: Read value of n

Step 3: Set sum=0 and i=1

Step 4: Calculate sum=sum+i

Step 5: Increment i

Step 6: Check i<=n

Step 7: Display the sum

Step 8: Stop

PROGRAM:

#include<stdio.h>

void main()

{

int n,sum=0;

i=l;

printf("Enter the limit\n");

scanf("%d",n);

do

{

sum=sum+i;

i++;

while(i<=n)

}

printf("sum of first n natural numbers is %d",sum);

}

OUTPUT:

Enter the limit

PROGRAM NO: 7

AIM: Construct a multiplication table using for loop

ALGORITHM

Step 1: Read the value of n,p,i

Step 2: Set i=0

Step 3: Calculate p=i\*n

Step 4: Display i,n,p

Step 5: Stop

PROGRAM

#include<stdio.h>

main()

{

int n,p,i;

printf("Enter a number\n");

scanf("%d",&n);

for(i=0;i<=10;i++)

{

p=i\*n;

printf("%d\*%d=%d\n",i,n,p);

}

}

OUTPUT

Enter a number

5

0\*5=0

1\*5=5

2\*5=10

3\*5=15

4\*5=20

5\*5=25

6\*5=30

7\*5=35

8\*5=40

9\*5=45

10\*5=50

PROGRAM NO: 8

AIM: Implementing the break and continue statement.

ALGORITHM Step 1 : start

Step 2 : read value of n from user

Step 3 : set i equal to 0 Step 4 : if i greater than n then go to step 5 else go to step 11

Step 5 : read value of a from user Step 6 : if a equal to 9999 go to step 11 else go to step 7

Step 7 : if a greater than 0 go to step 10 else go to step 8

Step 8 : calculate s equal to sqrt(a)

Step 9 : print value of a and s

Step 10 : i equal to i+1 go to step 4

Step 11 : stop

PROGRAM

#include<stdio.h>

#include<math.h>

void main()

{

int n,a,i;

float s;

printf("Enter the limit\n");

scanf("%d",&n);

printf("Enter the numbers\n");

for(i=0;i<n;i++)

{

scanf(“%d”,&a);

if(a==9999)

{

break;

}

if(a<0)

{

continue;

}

s=sqrt(a);

printf(“Square root of %d is %f\n”,a,s);

}

}

OUTPUT:

Enter the limit

2

Enter the numbers

9999

Enter the limit

2

Enter the numbers

16

Square root of 16 is 4.000000

81

Square root of 81 is 9.000000

PROGRAM NO: 9

AIM: Average of n numbers using goto statement.

ALGORITHM

Step 1 : start

Step 2 : read the value of n from user

Step 3 : set i equal to 0 Step 4 : if i greater than n then go to step 5 else go to step 9

Step 5 : read the value of p from user

Step 6 : if p less than 0 go to step 9 else go to step 7

Step 7 : calculate s equal to s+p

Step 8 : i equal to i + 1 go to step

4 Step 9 : calculate avg equal to (s/(i))

Step 10 : print the value of avg

Step 11 : stop

PROGRAM

#include<stdio.h>

void main()

{

int n,p,i,s=0;

float avg;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter the elements\n");

for(i=0;i<n;i++)

{

scanf(“%d”,&p);

if(p<0)

{

goto label;

}

s=s+p;

}

label:

avg=(s/(i));

printf(“Average=%f”,avg);

}

OUTPUT:

Enter limit

6

Enter the elements

3

4

6

8

9

10

Average=6.666666667

PROGRAM NO: 13

AIM: Find sum and average of numbers in an array.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0,s equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 go to step 4

Step 7: set s equal to call function fun()

Step 8: print the value of s

Step 9: stop

Function fun()

Step 1: start

Step 2: set i equal to 0

Step 3: if i less than n then go to step 6 else go to step 9

Step 4: calculate sum equal to sum+a[i]

Step 5: i equal to i+1 go to step 3

Step 6: Print value of sum

Step 7: calculate sum divide by n and value assign to avg

Step 8: return value avg

Step 7: stop

PROGRAM

#include<stdio.h>

float fun();

int a[25],i,n;

void main()

{

float s;

printf("Enter a limit");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

s=fun();

printf("Avg=%f",s);

}

float fun()

{

int sum=0;

float avg=0;

for(i=0;i<n;i++)

{

sum=sum+a[i];

}

printf("Sum=%d\n",sum);

avg=sum/n;

return avg;

}

OUTPUT:

Enter limit

4

Enter numbers

32

23

76

10

Sum=141

Avg=35.25

PROGRAM NO: 14

AIM: Largest and smallest element in an array.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function fun()

Step 8: stop

Function fun()

Step 1: start

Step 2: set large equal to 0

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 10

Step 5: if a[i] greater than large step 6 else got to step 7

Step 6: set large equal to a[i]

Step 7: if a[i] less than small go to step 8 else got to step 9

Step 8: set s equal to a[i]

Step 9: i equal to i+1 go to step 4

Step 10: print the value of large

Step 11: print the value of small

Step 12: stop

PROGRAM

#include<stdio.h>

void fun();

int a[10],n,i;

void main()

{

printf("Enter limit:\n");

scanf("%d",&n);

printf("Enter numbers:\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

fun();

}

void fun()

{

int large=0,small;

for(i=0;i<n;i++)

{

if(a[i]>large)

{

large=a[i];

}

if(a[i]<small)

{

small=a[i];

}

}

printf("Largest number is=%d\n",large);

printf("Smallest number is=%d\n",small);

}

OUTPUT:

Enter limit:

4

Enter numbers:

1

15

83

8

Largest number is=83

Smallest number is=1

PROGRAM NO: 15

AIM: Linear search using function.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function search ()

Step 8: if f equal to 0 print element not found

Step 9: Stop

Function search()

Step 1: start

Step 2: read the value of s from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 8

Step 5: if a[i] equal to s then go to step 6 else go to step 7

Step 6: print the value of i+1 go to step 10

Step 7: Set flag value equal to 1

Step 8: i equal to i+1 got to step 4

Step 9: Return value of f

Step 10: stop

PROGRAM

#include<stdio.h>

int search();

int f=0,n,a[25],p,i,l;

void main()

{

printf("Enter the limit\n");

scanf("%d",&n);

printf("enter the numbers\n");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

search();

if(f==0)

printf("not found\n");

}

int search()

{

int s;

printf("enter the number to be searched\n");

scanf("%d",&s);

for(i=0;i<n;i++)

{

if(a[i]==s)

{

printf("number found at %d position\n",i+1);

f=1; }

}

return f;

}

OUTPUT:

Enter the limit 5

enter the numbers

8

9

4

5

2

enter the number to be searched

7

not found

PROGRAM NO: 16

AIM: Selection sort using function

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function selection (a,n)

Step 8: stop

Function selection(a[ ],n)

Step 1: start Step 2: set i equal to 0

Step 3: if i less than n then go to step 3 else go to step 12

Step 4: set j equal to i+1

Step 5: if j less than n-1 then go to step 6 else go to step 11

Step 6: if ( a[i] > a[j] ) go to step 7 else got to step 10

Step 7: set temp equal to a[i]

Step 8: set a[i] equal to a[j]

Step 9: set a[j] equal to temp

Step 10: j equal to j+1 go to step 5

Step 11: i equal to i+1 go to step 3

Step 12: set i equal to 0

Step 13: if i less than n then go to step 14 else go to step 16

Step 14: print the value of a[i]

Step 15: i equal to i+1 got to step 13

Step 16: stop

PROGRAM

#include<stdio.h>

void selection(int [],int);

int i,j;

void main()

{

int a[10],n;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

selection(a,n);

}

void selection(int a[],int n)

{

int temp;

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("sorted numbers are\n"); for(i=0;i<n;i++) {

printf(" %d\n",a[i]);

}

}

38

OUTPUT:

Enter limit

5

Enter numbers

34

45

7

2

13

sorted numbers are

2

7

13

34

45

PROGRAM NO: 17

AIM: Bubble sort using function.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function bubble(a,n)

Step 8: stop

Function bubble(a[ ],n)

Step 1 : start

Step 2 : set i equal to 0

Step 3 : if i less than n then go to step 3 else go to step 12

Step 4 : set j equal to 0

Step 5 : if j less than n-i-1then go to step 6 else go to step 11

Step 6 : if ( a[j] > a[j+1] ) go to step 7 else got to step 10

Step 7 : set temp equal to a[j]

Step 8 : set a[j] equal to a[j+1]

Step 9 : set a[j+1] equal to temp

Step 10: j equal to j+1 go to step 5

Step 11: i equal to i+1 go to step 3

Step 12: set i equal to 0

Step 13: if i less than n then go to step 14 else go to step 16

Step 14: print the value of a[i]

Step 15: i equal to i+1 got to step 13

Step 16: stop

PROGRAM

#include<stdio.h>

void bubble(int [],int);

int i,j;

void main()

{

int a[10],n;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

bubble(a,n);

}

void bubble(int a[],int n )

{

int temp;

for(i=0;i<n;i++)

{

for(j=0;j<(n-i-1);j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

printf("sorted numbers are\n");

for(i=0;i<n;i++)

{

printf(" %d\n",a[i]);

}

}

OUTPUT:

Enter limit

5

Enter numbers

23

7

18

1

54

sorted numbers are

1

7

18

23

54

PROGRAM NO: 18

AIM: Matrix multiplication using function.

ALGORITHM

Step 1 : start

Step 2 : read the value of r1 and c1 from user

Step 3 : read the value of r2 and c2 from user

Step 4 : if r1 not equal to c2 then print multiplication not possible else go tostep 5

Step 5 : set i equal to 0

Step 6 : if i less than r2 then go to step 7 else go to step 12

Step 7 : set j equal to 0

Step 8 : if j less than c2 then go to step 9 else go to step 11

Step 9 : read the value of a[i][j] from user

Step 10: j equal to j+1 got to step 8

Step 11: i equal to i+1 go to step 6

Step 12: set i equal to 0

Step 13: if i less than r1 then go to step 14 else go to step 19

Step 14: set j equal to 0

Step 15: if j less than c2 then go to step 16 else go to step 18

Step 16: read the value of b[i][j] from user

Step 17: j equal to j+1 got to step 15

Step 18: i equal to i+1 go to step 13

Step 19: Print two metrices a[i][j] and b[i][j]

Step 20: Call function mul(a,b,r1,c2)

Step 21: stop

Function mul()

Step 1 : start

Step 2 : set i equal to 0

Step 3 : if i less than r1 then go to step 4 else go to step 1

3 Step 4 : set j equal to 0 43

Step 5 : if j less than c2 then go to step 6 else go to step 8

Step 6 : set m[i][j] equal to 0

Step 7 : set k equal to 0

Step 8 : if k less than r1 then go to step 9 else go to step 1

1 Step 9 : calculate m[i][j] equal to m[i][j]+(a[i][k]\*b[k][j]) S

tep 10: k equal to k+1 go to step 8

Step 11: j equal to j+1 go to step 5

Step 12: i equal to i+1 go to step 3

Step 13: set i equal to 0

Step 14: if i less than m then go to step 15 else go to step 20

Step 15: set j equal to 0

Step 16: if j less than n then go to step 17 else go to step 19

Step 17: print the value of m[i][j] Step 18: j equal to j+1 go to step 16

Step 19: i equal to i+1 go to step 14

Step 20: stop

PROGRAM

#include<stdio.h>

void mul(int [][10],int [][10],int ,int);

inti,j,r1,c1,r2,c2,k;

void main()

{

int a[10][10],b[10][10];

printf("Enter row and column of first matrix\n");

scanf("%d%d",&r1,&c1);

printf("Enter row and column of second matrix\n");

scanf("%d%d",&r2,&c2);

if(r1!=c2)

printf("not possible\n");

else

{

printf("enter the elements of first matrix\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("enter the elements of second matrix\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

scanf("%d",&b[i][j]);

}

}

printf("FIRST MATRIX\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("SECOND MATRIX\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

mul(a,b,r1,c2);

}

}

void mul(int a[][10],int b[][10],int r1,int c2)

{

int m[10][10];

printf("multiplication table\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

m[i][j]=0;

for(k=0;k<r1;k++)

{

m[i][j]=m[i][j]+(a[i][k]\*b[k][j]);

}

printf("%d\t",m[i][j]);

}

printf("\n");

}

}

OUTPUT

Enter row and column of first matrix

2

2

Enter row and column of second matrix

2

2

enter the elements of first matrix

1

3

2

3

enter the elements of second matrix

3

5

2

7

FIRST MATRIX

1 3

2 3

SECOND MATRIX

3 5

2 7

multiplication table

9 26

12 31

PROGRAM NO: 19

AIM: Sum of diagonal elements of a matrix.

ALGORITHM Step 1 : start

Step 2 : read the value of r and c from user

Step 3 :if r not equal to c then print not possible else goto step 4

Step 4 : set i equal to 0

Step 5 : if i less than r then go to step 6 else go to step 11

Step 6 : set j equal to 0

Step 7 : if j less than c then go to step 8 else go to step 10

Step 8 : read the value of a[i][j] from user

Step 9 : j equal to j+1 got to step 7

Step 10: i equal to i+1 go to step 5

Step 11: Print two metrix a[i][j]

Step 12: Call function dioganall(a,r,c)

Step 13: stop

Function diagonal()

Step 1 : start

Step 2 : set sum1 and sum2 equal to 0

Step 3 : set i equal to 0

Step 4 : if i less than r then go to step 5 else go to step 13

Step 5 : set j equal to 0

Step 6 : if j less than c2 then go to step 7 else go to step 8

Step 7 : if I equal to j then go to step 8

Step 8 : Find sum1+a[i][j] and store to sum1

Step 9 : if (i+j) equal to (r-1) then goto step 10

Step 10: find sum2+a[i][j] and store to sum2

Step 11: j equal to j+1 got to step 6

Step 12: i equal to i+1 go to step 4

Step 13: print the value of sum1 as sum of trace or principal diagonal elements

Step 14: print the value of sum2 as sum of diagonal elements

Step 15:stop

PROGRAM

#include<stdio.h>

void dioganal(int [][10],int,int);

inti,j;

void main()

{

int a[10],r,c;

printf("Enter row and column of matrix\n");

scanf("%d%d",&r,&c);

if(r!=c)

printf("Not possible\n");

else

printf("enter the elements of matrix\n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("matrix is\n");

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

dioganal(a,r,c);

}

void dioganal(int a[][10],int r,int c)

{

int sum1=0,sum2=0;

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

{

if(i==j)

49

{

sum1=sum1+a[i][j];

}

if((i+j)==(r-1))

{

sum2=sum2+a[i][j];

} }

}

printf("Sum of Trace or principl dioganal elements is=%d\n",sum1);

printf("Sum of off dioganal elements= %d\n",sum2);

}OUTPUT:

Enter row and column of matrix

2

2

enter the elements of matrix

1

3

5

4

matrix is

1 3

5 4

Sum of Trace or principl dioganal elements is=5

Sum of off dioganal elements= 5

Enter row and column of matrix

3

2

Not possible

PROGRAM NO: 20

AIM: String manipulations.

ALGORITHM

Step 1 : start

Step 2 : read the value of s from user

Step 3 : find the length of string s then go to step 4

Step 4 : print the string length

Step 5 : read value of s1 and s2 from user

Step 6 : copy the string s2 to s1

Step 7 :print the value of s1

Step 8 : read value of s3 and s4from user

Step 9 : compare s3 and s4 if it is equal to 0 then go to step 10 else go to step 11

Step 10 : print „strings are equal‟ Step 11 : print „strings are not equal

Step 12 : concatenate strings s3 and s4

Step 13 : print value of s3

Step 14: stop

PROGRAM

#include<stdio.h>

#include<string.h>

void main()

{

char s[20],s1[10],s2[10],s3[10],s4[10];

int n,h; printf("Enter the string:\n");

scanf("%s",s); n=strlen(s);

printf("\*\*\*\*LENGTH\*\*\*\*\n");

printf("Length of string using strlen: %d\n",n);

printf("Enter two strings:\n");

scanf("%s%s",s1,s2); strcpy(s1,s2);

printf("\*\*\*\*\*\*COPY\*\*\*\*\*\n");

printf("Copied string is:%s\n",s1);

printf("Enter two strings:\n");

scanf("%s%s",s3,s4); printf("\*\*\*\*\*COMPARE\*\*\*\*\n");

h=strcmp(s3,s4);

if(h==0)

{

printf("Two strings are equal\n");

}

else

{

printf("Two strings are not equal\n");

}

strcat(s3,s4);

printf("\*\*\*\*\*CONCATENATION\*\*\*\*\n");

printf("The concatenated string:%s\n",s3);

}

OUTPUT:

Enter the string:

neena

\*\*\*\*LENGTH\*\*\*\*

Length of string using strlen: 5

Enter two strings:

my name

\*\*\*\*\*\*COPY\*\*\*\*\*

Copied string is:name

Enter two strings:

my name

\*\*\*\*\*COMPARE\*\*\*\*

Two strings are not equal

\*\*\*\*\*CONCATENATION\*\*\*\*

The concatenated string:myname

PROGRAM NO: 21

AIM: Find length of a string using pointer.

ALGORITHM

Step 1: start

Step 2: read the value of str from user

Step 3: set p equal to str

Step 4: if \*p not equal to ‟\0‟ then go to step 5 else go to s

Step 5: c equal to c+1 go to step 6

Step 6: p equal to p+1 go to step 4

Step 7: print value of c

Step 8: stop

PROGRAM

#include<stdio.h>

void main()

{

char str[20],\*p;

int c=0;

printf("Enter a string\n");

scanf("%s",str);

for(p=str;\*p!='\0';p++)

{

c++;

}

printf("Length=%d\n",c);

}

OUTPUT:

Enter a string

college

Length=7

PROGRAM NO: 22

AIM: Sort list of names using array of pointer.

ALGORITHM

Step 1 : start

Step 2 : read the value of n from user

Step 3 : set i equal to 0

Step 4 : if i less than n then go to step 5 else go to step 10

Step 5 : read the value of s[i] from user

Step 6 : i equal to i+1 got to step 4

Step 7 : set i equal to 0

Step 8 : if i less than n then go to step 9 else go to step 16

Step 9 : set j equal to i+1

Step 10: if j less than n then go to step 11 else go to step 15

Step 11: compare &s[i ]and &s[j] if it is greater than 0 then go to step 12 else go to step15

Step 12: set temp equal to s[i]

Step 13: set s[i] equal to s[j]

Step 14: set s[j] equal to temp

Step 15: j equal to j+1 go to step 10

Step 16: i equal to i+1 go to step 8

Step 17: print the value of s[i]

Step 18: stop

PROGRAM

#include<stdio.h>

void main()

{

char \*s[20],\*temp;

int i,j,n;

printf("Enter limits:\n");

scanf("%d",&n);

printf("Enter names:\n");

for(i=0;i<n;i++)

{

scanf("%s",&s[i]);

//gets(s[i]);

}

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(strcmp(&s[i],&s[j])>0)

{

temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}

}

printf("Sorted names are:\n");

for(i=0;i<n;i++)

{

printf("%s\n",&s[i]);

}

}

OUTPUT

Enter limits:

5

Enter names:

Anna

Varun

Sara

Alan

Jis

Sorted names are:

Alan

Anna

Jis

Sara

Varun

PROGRAM NO: 23

AIM: Implementation of pointer to pointer

ALGORITHM

Step 1 : start

Step 2 : set p1 equal to &a

Step 3 : set p2 equal to &p1

Step 4 : set q1 equal to &b

Step 5 : set q2 equal to &q1

Step 7 : read the value of a and b

Step 8 : calculate c equal to \*\*p2 + \*\*q2

Step 9 : print the value of c

Step 10 : calculate c equal to \*\*p2 - \*\*q2

Step 11 : print the value of c

Step 12 : calculate c equal to \*\*p2 \* \*\*q2

Step 13 : print the value of c

Step 14 : calculate c equal to \*\*p2 / \*\*q2

Step 15 : print the value of c

Step 16 : stop

PROGRAM

#include<stdio.h>

void main()

{

int a,b,c,\*p1,\*\*p2,\*q1,\*\*q2;

p1=&a;

p2=&p1;

q1=&b;

q2=&q1;

printf("Enter two numbers\n");

scanf("%d%d",&a,&b);

c=\*\*p2+\*\*q2; 57 printf("sum=%d\n",c);

c=\*\*p2-\*\*q2; printf("Difference=%d\n",c);

c=\*\*p2\*\*\*q2; printf("product=%d\n",c);

c=\*\*p2/ \*\*q2; printf("division=%d\n",c);

}

OUTPUT:

Enter two numbers

8

4

sum=12

Difference=4

product=32

division=2